



Programación de un terminal satélite Inmarsat

Proyecto fin de carrera

Tutor pedagógico : René BOULAIRE

Carmen LÓPEZ LIÉBANAS

Autora: Carmen López Liébanas

Tutor pedagógico: René Boulaire

Institución en la que se ha realizado: Ecole des Arts et Métiers

Coordinador Académico del convenio: Elisa M^a Ruiz Navas

Cotutor en la UCIIM: Elisa M^a Ruiz Navas

Fecha de lectura: 8 febrero 2012

Programación de un terminal satélite Inmarsat

Proyecto fin de carrera

Contenido

I.OBJECTIVOS.....	4
II.INTRODUCCIÓN	5
1. SkyWave	5
2. Réseaux.....	Erreur ! Signet non défini.
3. Terminal SureLinx 8100c.....	5
4. Programas.....	6
a. Scope.....	6
b. Message Designer	6
c. Conecte	6
5. Ejemplo.....	6
III.MICROCONTROLADOR.....	8
IV.SOLUCIÓN	9
1. Recordatorio de objetivos	9
2. NMEA	9
a. Generador de tramas NMEA.....	9
3. Programme	10
a. Programa principal	10
b. Administrador de bytes recibidos	11
c. Administrador de tramas recibidas	11
d. Tratamiento de la trama	12
e. Guardar los valores en un espacio de memoria	12
g. Ejemplo.....	13
h. Résultat.....	13
4. Message Designer	14
V.CONCLUSIÓN	15
VI.BIBLIOGRAFÍA	16

I. OBJETIVOS

En el ámbito del transporte viario, ferroviario o marítimo, para la gestión de flota puede ser interesante obtener información de los vehículos móviles que ellos mismos gestionan. Esta información puede ir desde la localización del barco hasta el estado de funcionamiento del vehículo pasando por otras informaciones diversas como el nivel de gasolina, las condiciones meteorológicas, el desplazamiento u otros datos que nos pueden aportar los sensores pero que están inaccesibles a causa de la distancia y de la movilidad en los barcos.

TrackWave es una empresa que aporta soluciones de seguimiento marítimo (barcos de pesca, de placer y de carreras) y terrestre. Trabajan con la empresa SkyWave y necesitan una aplicación marítima que reconozca las normas marítimas.

Para poner a disposición la información del vehículo móvil a un receptor fijo, es decir, el cliente final, se necesita un aparato que permite la comunicación con dicho vehículo. Ello permite la realización de soluciones de seguimiento, vigilancia y control.

Existen muchas tecnologías para enviar la información a distancia, la más eficaz es a través de un satélite porque permite vínculos estables y eficaces de forma continua con respecto a la radio.

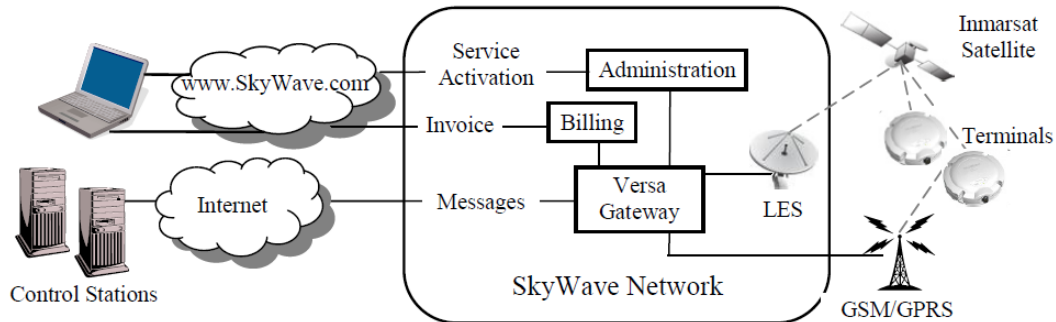
SkyWave fabrica no solo el terminal, pero también toda la infraestructura de telecomunicación necesaria para la comunicación, y un conjunto de programas que nos permitirán personalizar nuestra instalación para adaptarlo a nuestras necesidades y a las de la empresa TrackWave. SkyWave es la única empresa que permite la comunicación por GPRS y por satélite, y lleva integrado un microcontrolador programable en C el cual permite concebir soluciones más complejas pero sobre todo más adaptadas.

II. INTRODUCCIÓN

1. SkyWave

SkyWave Mobile Communications es un proveedor mundial de equipos de comunicación por satélite para aplicaciones máquina a máquina. Fue fundado en 1997, en Ottawa, Canadá. Han vendido más de 400 000 terminales en más de 75 países. Podemos aplicar sus productos en el transporte viario, ferroviario, marítimo, en el sector del petróleo y del gas y también para el gobierno y la defensa.

2. Redes



Los mensajes enviados pasan por la pasarela Versa SkyWave hasta la estación de control. Los terminales pueden estar en movimiento o pueden estar fijos.

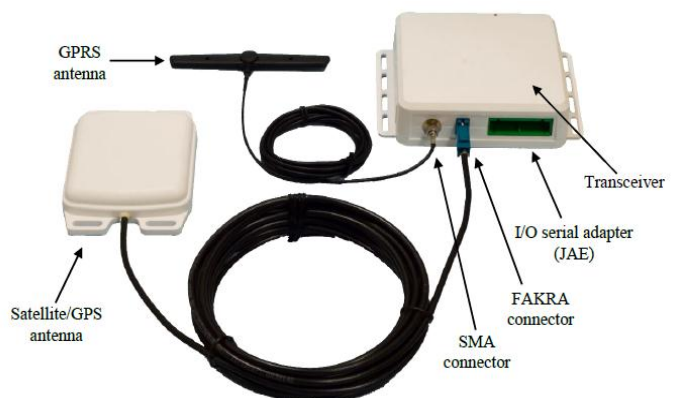
Explicación de las diversas partes de la red Versa:

- **La Estación de Control del Proveedor de Soluciones:** El proveedor de Soluciones gestiona todos los aspectos de la interfaz para el usuario final. Cada estación de control tiene un identificador.
- **Pasarela Versa:** Permite la comunicación sin cables con los terminales por todo el mundo utilizando los satélites Inmarsat o el servicio terrestre GPRS.
- **Servicio de datos satélite SkyWave :** SkyWave satélite es un sistema de comunicación en los dos sentidos, semi-duplex con una cobertura global basada en la norma Inmarsat D+. Ofrece una tasa de datos relativamente baja, un tiempo casi real de transferencia de datos y un coste de mensaje bajo.
- **Servicio de datos terrestre por GPRS:** GPRS (General Packet Radio Service) es el sistema de comunicación más expandido del mundo, consiste en una comunicación basada en los paquetes de datos, donde se pueden enviar informaciones vía la red GSM. Está orientado a la transmisión de datos. La cobertura GPRS se concentra generalmente en las zonas pobladas como las ciudades y las carreteras principales, es por este motivo por el que la comunicación por GPRS no es interesante para nuestro proyecto.
- **Terminales:** SkyWave posee una gama de productos con distintas aplicaciones.

La comunicación entre los terminales y el servidor puede llevarse a cabo por la pasarela Versa o con GPRS utilizando un programa intermediario llamado Conecte que establece una unión directa a la Estación de Control.

3. Terminal SureLinx 8100c

SureLinx es una unidad autónoma, incluyendo un transmisor-receptor de satélite (módem), un modem GPRS, un módulo GPS, un controlador programable en C y múltiples I/O (entradas/salidas) capaces de controlar y vigilar los sensores y aparatos externos. La unidad principal ha sido diseñada para ser situada en el interior de un vehículo.

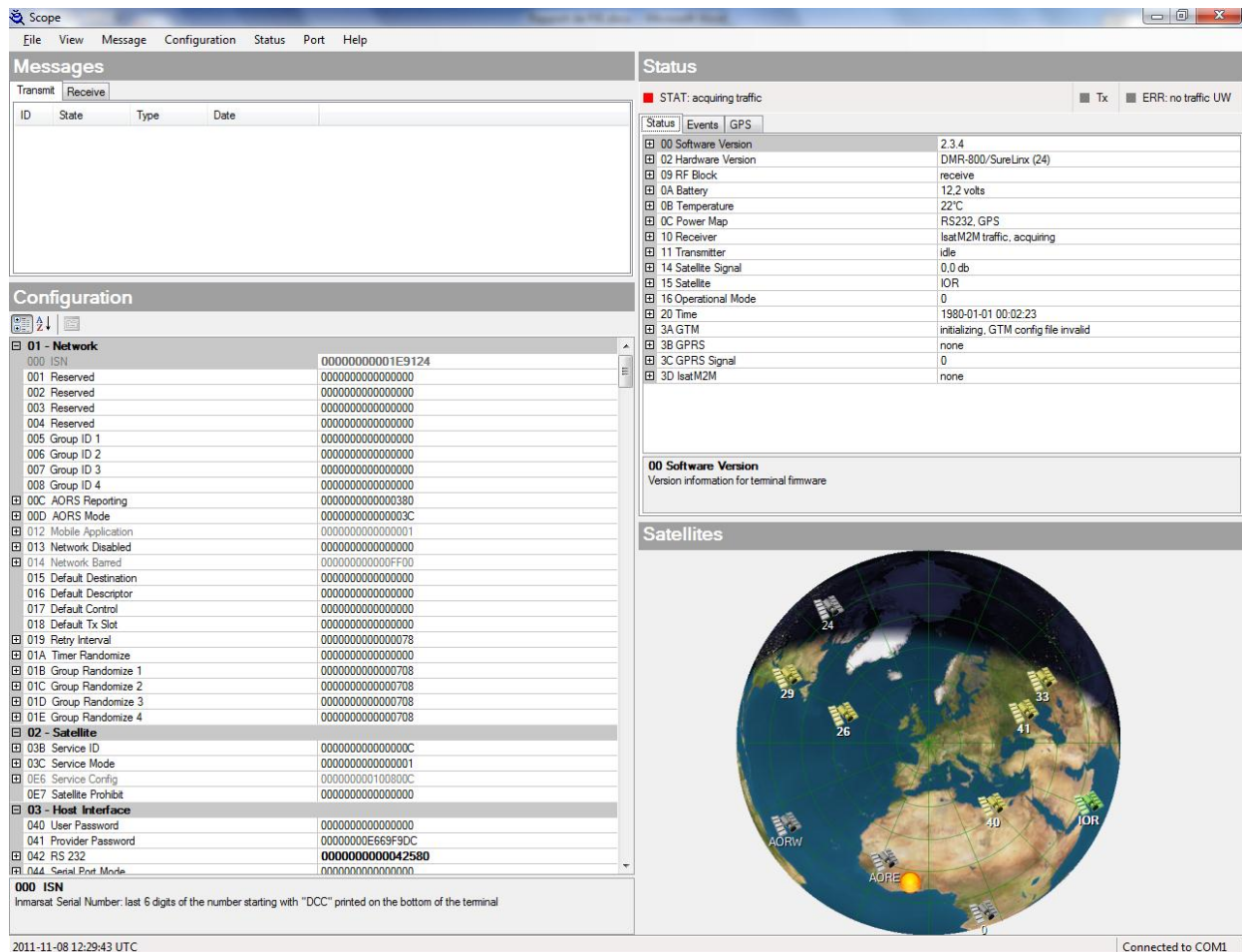


4. Programas

La empresa aporta un gran abanico de programas de base que permiten la configuración y la programación del terminal. Podemos crear un programa en lenguaje C pero siempre tendremos que utilizar estos programas para completar los objetivos. Existen muchos programas pero los más utilizados e importantes para nuestro proyecto son los siguientes:

a. Scope

Scope proporciona una interfaz gráfica para los terminales SkyWave a través de una interfaz RS-232. Gracias a este programa podemos ver el comportamiento del terminal. Con él, podemos controlar, ensayar, ver y configurar el terminal.



b. Message Designer

Message Designer es un programa que permite crear mensajes personalizados llamados *poll*. Estos mensajes son de tamaño fijo, de costes fijos y de latencia previsible.

c. Conecte

Conecte es un servicio integrado de colecta de mensajes que recibe a la vez mensajes provenientes de satélites o por GPRS. Ha sido concebido para ser utilizado como una base de un servidor de producción de mensajes.

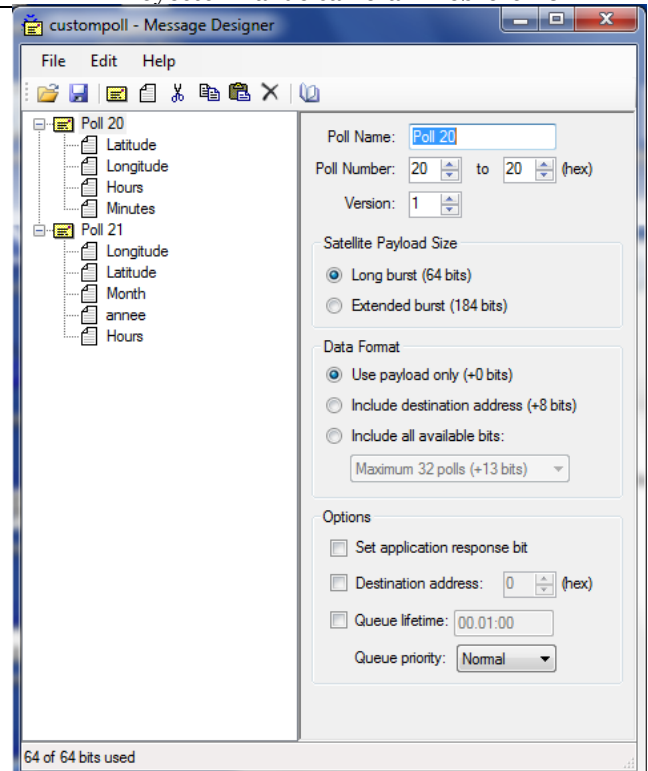
5. Ejemplo

Para nuestro proyecto hemos utilizado un ordenador, un servidor con una base de datos instalada en él, un terminal y un generador de tensión. El ordenador está unido al terminal por el puerto serie del ordenador. A continuación, el terminal es alimentado por el generador. El terminal puede enviar la información al servidor por satélite o por GPRS. Para facilitar el uso del servidor se puede utilizar el programa Conecte del que ya hablamos en el apartado anterior.

Para crear un poll personalizado, hay que utilizar el programa Message Designer para personalizar los campos del poll. Hemos creado nuestro propio poll, llamado poll 20. Nos da la latitud, la longitud, la hora y los minutos en el momento del envío.

Asignamos el espacio de memoria para cada campo. Para la latitud, podemos tener valores comprendidos entre -90° et $+90^{\circ}$, luego necesitamos 2 cifras significativas antes de la coma. Por el contrario, para la longitud, los datos son entre -180° et 180° , luego necesitaremos 3 cifras significativas antes de la coma. Para ser precisos, hemos decidido que queremos una precisión de 10 metros.

Una vez enviado el mensaje, podemos ver el resultado en la base de datos gracias al programa Conecte. Los mensajes pasan a través del Satélite Inmarsat y a través de la pasarela Versa.



Return Messages										
Message Type: Poll_20 (Poll 20)										
Terminal	ReceiveTime	LogDateTime	LogType	PollNumber	ReturnMessage_ID	Sequence	Latitude	Longitude	Hours	Minutes
4776281	26/10/2011 09:02			20	38	-1	48,83361	2,35833	8	59

III. MICROCONTROLADOR

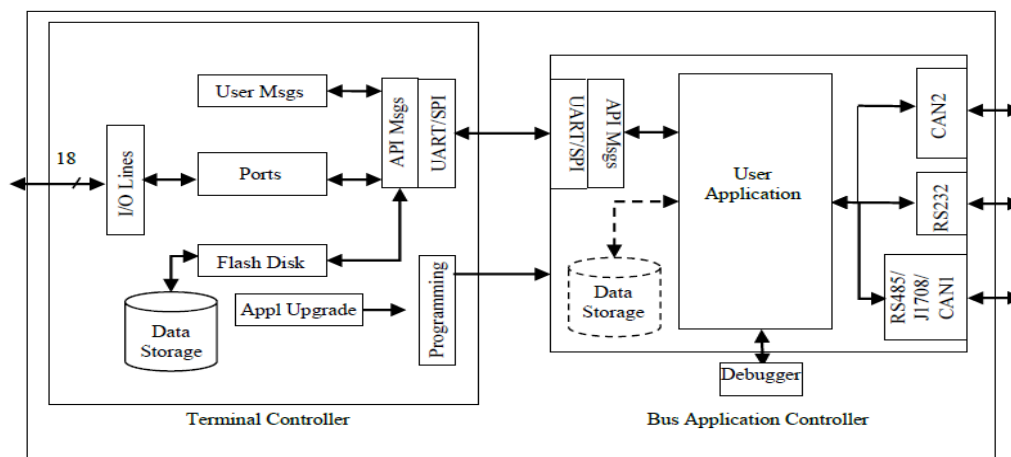
Un **microcontrolador** es un circuito integrado que incluye los elementos esenciales de un ordenador: procesador, memorias, periféricos e interfaces externas.

Una **interfaz de programación** (Application Programming Interface ou API) es una interfaz prevista para un programa informático. Permite la interacción de programas unos con otros y hace posible la interacción del hombre con la máquina.

SureLinX 8100c ofrece un entorno de concepción de aplicación mas potente proporcionando un microcontrolador embarcado para tratar las tareas complejas.

La figura 1 muestra como el microcontrolador se conecta al terminal. Los dos controladores se comunican el uno con el otro con el interfaz de programación API a través del interfaz de comunicación SPI utilizando un enlace en serie directo.

Figure 1 Terminal Controller and the Bus Application Controller



El microcontrolador dispone de un conjunto de instrucciones optimizado para el compilador en C, programación del puerto serie, memoria flash y EEPROM, acelerómetros MEMS de 3 ejes.

El terminal y el microcontrolador poseen distintos tipos de memorias:

- Una memoria EEPROM de 4 MB.
- Una mémoire flash de 2 MB.
- Un registro llamado Scratchpad de 64 puertos de 32 bits.

Después de hacer comparado las 3 memorias, hemos elegido el registro Scratchpad porque permite el acceso a la vez del terminal y del microcontrolador. En esta memoria solo podemos guardar los datos de tipo WORD32, que son unsigned long int (32 bits).

IV. SOLUCIÓN

1. Recordatorio de objetivos

Nuestra solución tiene que permitir al terminal SureLinX 8100c comunicarse con los aparatos de navegación marítimos. La mayor parte de estos aparatos utilizan la norma NMEA descrita a continuación. Técnicamente nuestros objetivos son los siguientes:

1. Recibir los datos en el puerto serie del microcontrolador,
2. Aislar una trama completa,
3. Aislar cada campo de la trama,
4. Modificar los datos si es necesario,
5. Poner a disposición del terminal los datos para enviarlos con un mensaje poll ,
6. Decodificar los datos modificados para registrarlos en la base de datos.

2. NMEA

La norma NMEA 0183 es una especificación para la comunicación entre los equipos marítimos. Está definida y controlada por la *National Marine Electronics Association* (NMEA), asociación americana. El objetivo es estandarizar los equipos marinos. La norma es una simple **comunicación serie** para transmitir una frase a uno o varios oyentes. Todos los datos son transmitidos bajo la forma **de carácter ASCII**, a una velocidad de **4800 bauds**. Todas las tramas **empiezan por el carácter \$**.

Seguido por un grupo de dos letras para el identificador del receptor:

GP = GPS: Global Positioning System

VD = Sensor de velocidad efecto dopler

RA = Radar

VM = Sensor de velocidad magnético

Después **un grupo de tres letras** para el identificador de la trama:

APA: Autopiloto "A"

APB: Autopiloto "B"

GGA: Definido por los datos del GPS.

GLL: Posicionamiento geográfico Longitud-Latitud.

Vienen a continuación un cierto número de campos separados por comas. El papel de la coma es ser el separador de los campos, así permite la deconcatenación de estos.

Al final aparece un campo opcional llamado checksum precedido por el signo *, que representa la OR exclusiva de todos los caracteres comprendidos entre \$ y *

Seguido al final de la línea con un [CR] <Retorno de carro> y [LF] <Retorno a la línea>. Hay un total de 82 caracteres máximos en cada trama.

Ejemplo: Waypoint Arrival Alarm: \$GPAAM, A, A, 0.10, N, WPTNME*43[CR][LF]

AAM: Arrival alarm

A: Arrival circle entered

A: Perpendicular passed

0.10: Circle radius

N: Nautical miles

WPTNME: Waypoint name

***43:** Checksum data

a. Generador de tramas NMEA

En nuestro proyecto, el generador de tramas es utilizado para crear distintas tramas. Utilizamos una interfaz de salida NMEA de NKE. Genera 12 tramas diferentes con diversa información: Velocidad de superficie, el registro diario, el registro total, el rumbo, la hora en UTC, la fecha, la profundidad, la temperatura del agua,

la temperatura del aire, la presión atmosférica, la velocidad y el ángulo del viento aparente y del real, rumbo y distancia al Waypoint, la desviación del rumbo y la latitud y la longitud.

En el programa Scope podemos ver las 12 tramas.

Status	Events	GPS
2012-01-06 10:35:26	application channel 0	"\$IHD,M,087.,M*1D"
2012-01-06 10:35:26	application channel 0	"HDM"
2012-01-06 10:35:26	application channel 0	"\$IXDR,A,004.,D,mastangle*3B"
2012-01-06 10:35:26	application channel 0	"\$IDPT,0004.6.,*70"
2012-01-06 10:35:26	application channel 0	"\$IDBT,0015.1f,0004.6,M.,*7E"
2012-01-06 10:35:26	application channel 0	"\$IVLW,0019.0,N,019.01,N*4C"
2012-01-06 10:35:26	application channel 0	"\$IVHW,,,087.,M,04.88,N,09.04,K*19"
2012-01-06 10:35:26	application channel 0	"\$IVWR,020.,R,24.9,N,12.8,M,046.1,K*7C"
2012-01-06 10:35:26	application channel 0	"\$IMWD,,,086.,M,18.5,N,09.5,M*00"
2012-01-06 10:35:27	application channel 0	"\$IMTA,10.8,C*0C"
2012-01-06 10:35:27	application channel 0	"\$IMTW,8.0,C*2B"
2012-01-06 10:35:27	application channel 0	"\$IVWT,000.,R,18.5,N,09.5,M,034.3,K*7B"
2012-01-06 10:35:27	application channel 0	"\$IHDG,087.....*76"
2012-01-06 10:35:27	application channel 0	"\$IHD,M,087.,M*1D"

3. Programme

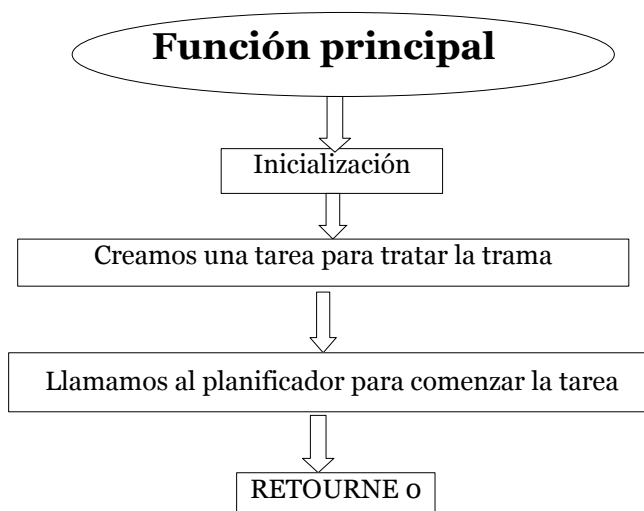
Un programa como este no ha sido nunca hecho por SkyWave. Este programa reúne la comunicación por serie y la programación en C para hacer el recorte de la trama, protocolo muy común en el transporte marítimo. Hemos sido nosotros los primeros que hemos elaborado todo el algoritmo y que hemos hecho la programación en lenguaje C.

El programa, en principio, hace una inicialización de todas las variables. Posteriormente, el microcontrolador recibe una trama NMEA carácter por carácter por el puerto serie. El programa reconoce de trama se trata y la comienza a tratar. Corta la trama quedándose con la información interesante y hace la conversión de los datos que lo necesitan. Después lo guarda en la memoria Scratchpad y lo podemos ver por pantalla para ver si el resultado es el correcto.

a. Programa principal

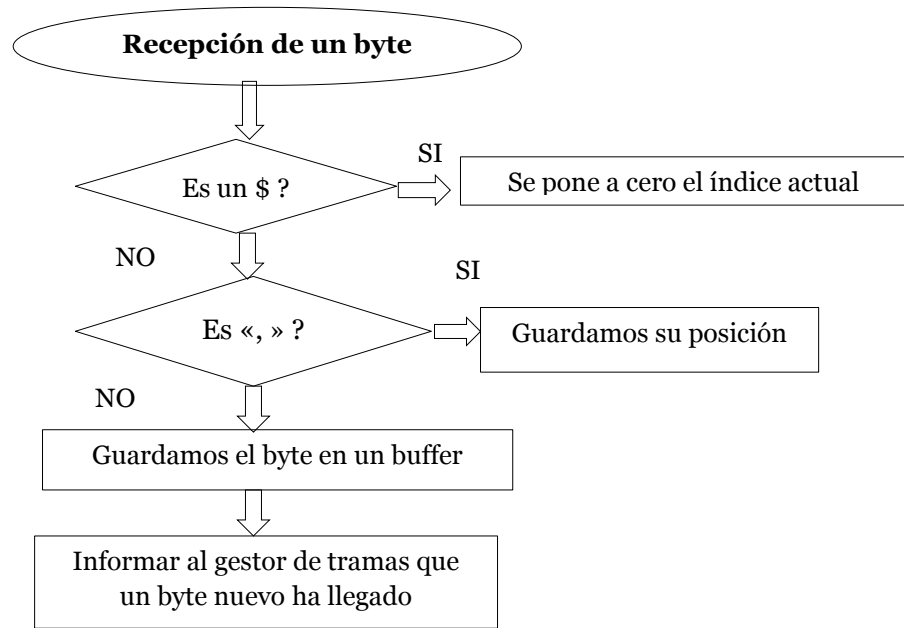
El programa principal inicializa el microcontrolador y el programa RTOS. También hace la inicialización de la comunicación en serie, es decir, la configuración del puerto serie, 4800 bauds es la tasa que respeta la norma NMEA.

A continuación, se crea una tarea. Una tarea es un término utilizado en los procesos de sistemas de explotación en tiempo real (RTOS) de sistemas embarcados. Cada tarea es un proceso independiente. RTOS es el encargado de gestionar las diferentes tareas. Con el Scheduler (planificador) hacemos que la tarea comience y con ello empieza el programa. Esto es útil si tenemos muchos cálculos que deben ser tratados al mismo tiempo y un único procesador.



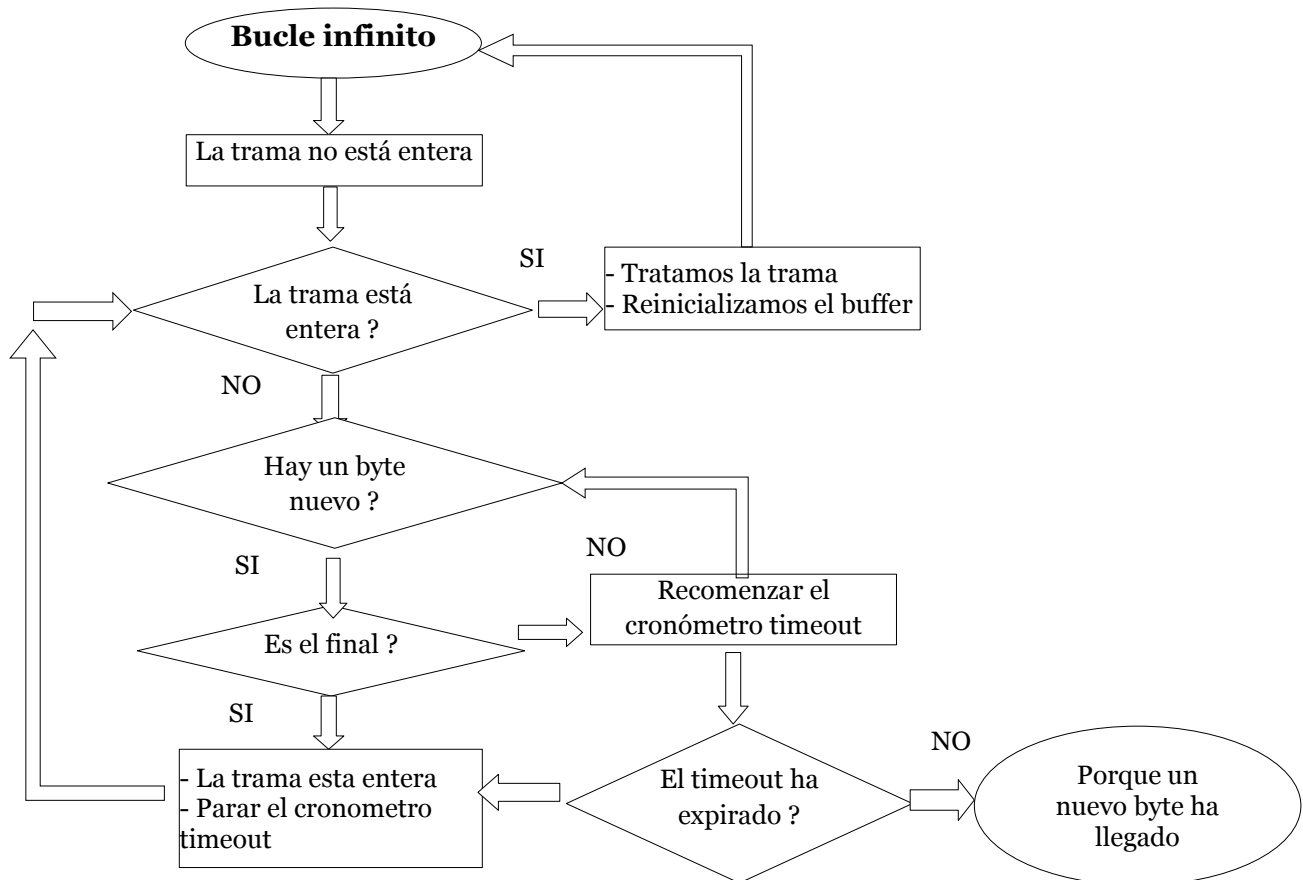
b. Administrador de bytes recibidos

Esta función comienza cuando hay valores en el puerto serie del microcontrolador. Si es el primer carácter, es decir, el símbolo \$, pone a cero el índice del momento para indicar que el primer valor de la trama. Si el carácter es un coma, guarda la posición de la coma. A continuación guarda cada byte en un buffer y avisa a la siguiente función que un nuevo byte ha llegado.



c. Administrador de tramas recibidas

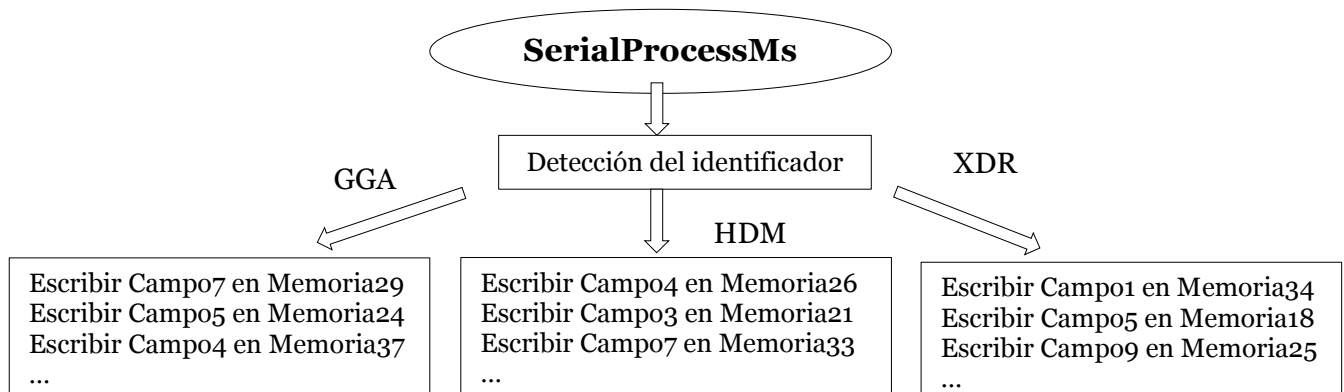
Esta función hace que podamos recibir una trama NMEA entera. Cuando llega un nuevo carácter comprueba si es el final de la trama o no. Si la trama esta completa, llamamos a la función que reconoce de que trama se trata. Si no esta completa, continuamos recibiendo bytes hasta completarla. Para ello nos ayudamos de un timer, si el timer finaliza y no ha llegado otro byte podemos decir que ha pasado demasiado tiempo y que la trama esta completa. Por lo tanto, en ese caso también llamamos a la función que identifica la trama.



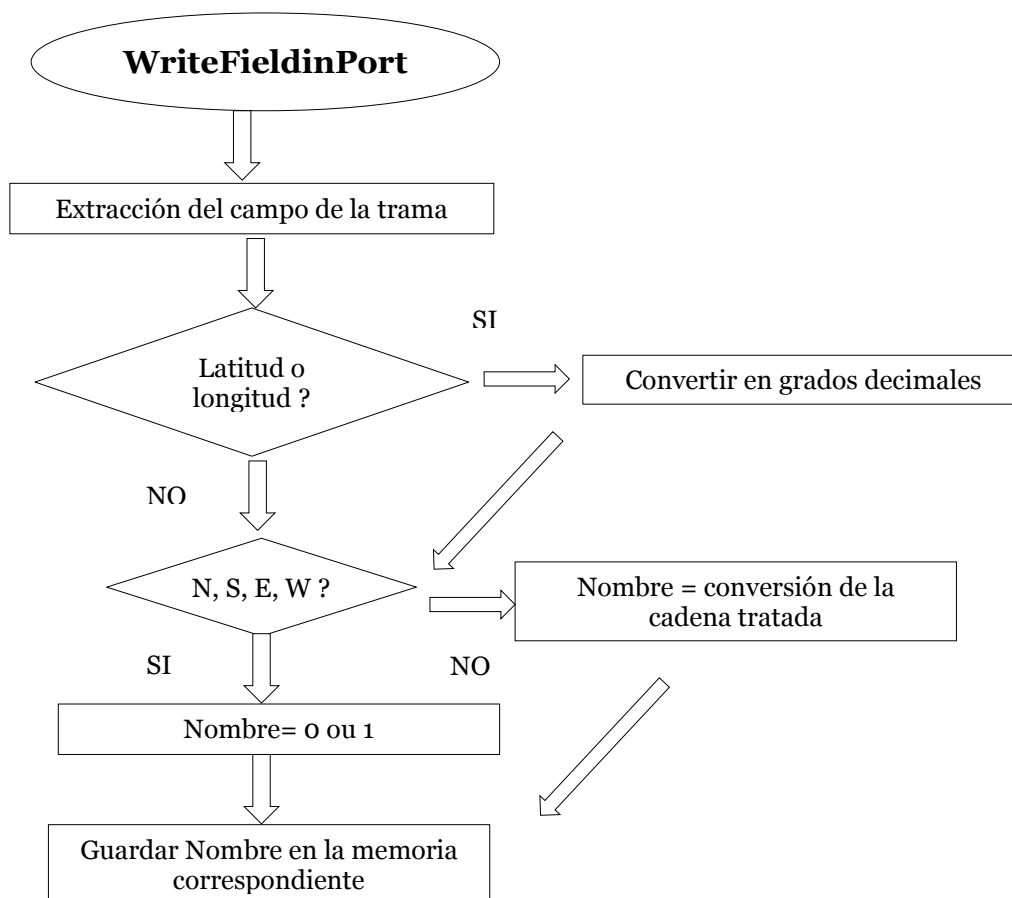
d. Tratamiento de la trama

Esta función comienza cuando una trama entera ha sido recibida. Su trabajo es tratar la trama.

Al principio, fijamos la trama completa en la pantalla para verificar que la trama ha llegado correctamente. A continuación, aislamos las 3 primeras letras que indican el identificador de la trama. En función del identificador, se va a llamar a la función `WriteFieldinPort` para ordenar la grabación de ciertos valores en distintas memorias. La función `WriteFieldinPort` recibe el número del campo (es decir, la posición del dato en la trama) y la dirección para guardarlo (Scratchpad).



e. Guardar los valores en un espacio de memoria



La table `indexComma` contiene las posiciones de las comas. Fue rellenado durante la recepción de los bytes uno por uno.

El primer argumento `fieldNumber` de la función `WriteFieldinPort` determina el campo que se quiere extraer. Gracias a la tabla `indexComma` se conoce la posición de las comas.

El segundo argumento indica el espacio de memoria donde serán guardados los datos. Como no se pueden guardar datos como números reales, lo que hace la función es eliminar la coma de la cadena. Para cada campo conocemos la precisión (numero de cifras después de la coma) luego podremos fácilmente dividir por la buena potencia de 10 al decodificar. Esta operación se llevará a cabo en la creación del poll.

f. Convertir la latitud y la longitud

Esta función convierte el valor de la latitud y la longitud de grados/minutos a grados decimales. Por ejemplo, de la trame recibimos el valor de la latitud 4807.038. 48 son los grados y 07.038 son los minutos. Para hacer la transformación dividimos 07.038 60.

$$\frac{07.038}{60} = 0.1173$$

Luego, la latitud en grados sera 48.1173°. Hacemos el mismo cálculo para la longitud, pero esta contiene una cifra mas en su parte entera. La latitud tiene 7 cifras y la longitud 8.

Latitud antes de la transformación:

field[8]							
4	8	0	7	.	0	3	8
ltg = 2		minutesByte[6]					

Latitud despues de la transformación :

field[8]							
4	8	1	1	7	3	0	0
ltg = 2		minutesByte[6]					

Se ha eliminado la coma para poderlo guardar en la memoria Scratchpad. Cuando hagamos el poll personalizado, recuperaremos la posición de la coma dividiendo por 1 000 000.

g. Ejemplo

Queremos guardar el valor de la altitud que se encuentra en la trama siguiente:

\$GPGGA,123519,4807.038,N,01131.324,E,1,08,0.9,545.4,M,46.9,M,,*42

Se trata del campo número 9, luego está entre la novena y decima coma. Después de la extracción del valor, obtenemos la cadena « 545.4 ».

Un bucle recorre todos los caracteres de la cadena para detectar el punto. Si un punto es encontrado, se reduce la talla del campo y desplazamos los caracteres una posición. Nuestra cadena después es « 5454 ».

Tenemos nuestro dato preparado para ser guardado en Scratchpad.

h. Résultat

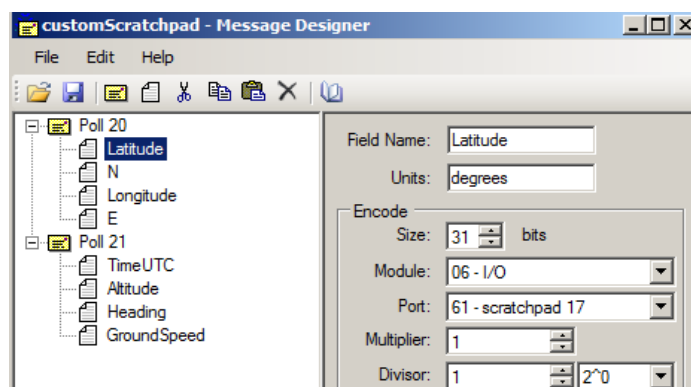
Este es el resultado final que podemos ver gracias al programa Scope.

Status	Events	GPS
2012-01-12 10:16:58	application channel 0	"Value read : 1"
2012-01-12 10:16:59	application channel 0	"Value read : 5454"
2012-01-12 10:17:03	application channel 0	"\$GPGGA,123519,4807.038,N,01131.324,E,1,08,0.9,545.4,M,46.9,M,,*42~"
2012-01-12 10:17:03	application channel 0	"GGA"
2012-01-12 10:17:03	application channel 0	"Value read : 123519"
2012-01-12 10:17:03	application channel 0	"Value read : 48117300"
2012-01-12 10:17:03	application channel 0	"Value read : 1"
2012-01-12 10:17:03	application channel 0	"Value read : 11522067"
2012-01-12 10:17:03	application channel 0	"Value read : 1"
2012-01-12 10:17:04	application channel 0	"Value read : 5454"
2012-01-12 10:17:04	receiver wide band	acquired
2012-01-12 10:17:08	application channel 0	"\$GPGGA,123519,4807.038,N,01131.324,E,1,08,0.9,545.4,M,46.9,M,,*42~"
2012-01-12 10:17:08	application channel 0	"GGA"

Podemos ver que los números son enteros, los valores que deberían ser decimal son todavía enteros. Es como los tenemos que guardar en el registro Scratchpad. Con el poll cambiaremos los valores que deban ser cambiados.

4. Message Designer

A continuación, para recibir los datos en la base de datos, es decir, el cliente final, creamos dos polls, Poll20 y Poll21 gracias al programa Message Designer. Rellenamos los campos necesarios y en el apartado de módulo tenemos que poner I/O y en el de puerto: Scratchpad correspondiendo a lo que definimos en nuestro programa en C.



Hacemos el cálculo del número de bits necesarios para cada tipo de dato. Este es el resultado final que podemos ver en el Servidor, con el programa Conecte.

Poll20

Latitude	N	Longitude	E
48,117308	<input checked="" type="checkbox"/>	11,522064	<input checked="" type="checkbox"/>
	<input type="checkbox"/>		<input type="checkbox"/>

Poll21

TimeUTC	Altitude	Heading	GroundSpeed
123519	545,4	0,0	0,0
123519	545,4	0,0	0,0

V. CONCLUSIÓN

Como ya he dicho, mi objetivo principal es enviar información proveniente de los barcos al gerente de flota. He finalizado una solución completa que pone a disposición del cliente los datos en una base. Se envía mensajes cortos y suficientes para nuestras necesidades. Luego, el terminal SkyWave 8100c se ajusta exactamente a los objetivos del proyecto.

En los barcos, muchos aparatos electrónicos se comunican utilizando la norma NMEA. **Propongo una posible solución para transmitir los datos de cualquier trama NMEA utilizando un poll.** Para hacer esto, necesitamos un programa en lenguaje C cargado en el microcontrolador. También un fichero de diseño de mensajes que permite decodificar los datos, es decir, encontrar el valor en número real a partir del número entero.

He conseguido decodificar las tramas NMEA que provienen de los aparatos electrónicos situados en los barcos. He creado un programa genérico en función de los aparatos de navegación. Es decir, el programa aísla los valores que pueden ser interesantes de cada trama NMEA y los trata para obtener el resultado conveniente.

Durante el proyecto trabajo en distintos campos. Primero, he instalado el Servidor de Windows y he creado una base de datos. A continuación, he utilizado la electrónica y la informática para conectar los aparatos electrónicos y hemos hecho ensayos con los programas para coger habilidad. Finalmente, he programado un microcontrolador y he conseguido que cumpla los objetivos.

VI. BIBLIOGRAFÍA

La bibliografía esta citada en orden de aparición en el texto.

Objetivos

- Empresa Trackwave : http://www.trackwave.fr/rubrique.php?id_rubrique=9
- Wikipedia Inmarsat : <http://en.wikipedia.org/wiki/Inmarsat>

Introducción

- Empresa Skywave : www.skywave.com
- Empresa Inmarsat : <http://www.inmarsat.com>
- DOCUMENTACIÓN SKYWAVE. - *N6_Versa_Network_Services_Guide*-, - *T10_DMR_800_Installation_and_Hardware_Guide*-, - *R4_Getting_Started_SureLinux_Evaluation*-, - *R1_Getting_Started_DMR-800_Evaluation*-, *AN19_Vehicle_Tracking*.
- <http://www.wolframalpha.com/input/?i=-89.55555+pasar+a+binary>
- <http://www.youtube.com/watch?v=IwCLRkqotQo>
- http://es.wikipedia.org/wiki/Sistema_binario

Microcontrolador

- DOCUMENTACIÓN SKYWAVE. - *T7_SureLinux_8100_Installation_and_Hardware_Guide*-, - *T8_DMR_800_and_SureLinux_Users_Guide*-, - *T9_DMR_800_and_SureLinux_API_Reference*-, - *T11_Application_Controller_API_Reference*-
- Microcontrolador: <http://fr.wikipedia.org/wiki/Microcontr%C3%B4leur>
- Interfaz de programación : http://fr.wikipedia.org/wiki/Interface_de_programmation
- Sistema de explotación en tiempo real :
 - ROSE Brique, -*Systèmes temps-réel embarqués*-, École Nationale Supérieure des Télécommunications.
 - THOMA Yann, -*Programmation Temps Réel. Introduction*-, Institut REDS, Haute Ecole d'Ingénierie et de Gestion – VD, Septembre 2010
 - NAVET Nicolas, -*Introduction aux Systèmes Temps Réel*-, 2010/2011, <http://www.loria.fr/~nnavet>
 - BOUZEFRANE Samia, -*Ordonnancement centralisé de tâches temps réel*-, Maître de Conférences, CEDRIC –CNAM, <http://cedric.cnam.fr/~bouzefra>
 - SADEG B., -*Systèmes temps réel et Ordonnancement*- Université du Havre - UFR ST - LITIS - Équipe STI, UFR des Sciences et Techniques

Solución

- NMEA :
 - <http://www.nmea.org/>
 - <http://fr.wikipedia.org/wiki/NMEA>
 - http://www.gpspassion.com/forumsen/topic.asp?TOPIC_ID=17661
 - <http://agirard.free.fr/GPS/NMEA.php>
 - http://www.voilelec.com/pages/can_nmea.php
 - grandzebu.net/informatique/utiles/nmea.rtf
 - http://en.wikipedia.org/wiki/NMEA_0183
 - <http://www.gpsinformation.org/dale/nmea.htm>
- YU Haobo, GERSTLAUER Andreas, GAJSKI Daniel, - *RTOS Scheduling in Transaction Level Models*-, CECS Technical Report 03-12, March 20, 2003, Center for Embedded Computer Systems, Information and Computer Science, University of California, Irvine. http://www.cecs.uci.edu/technical_report/TR03-12.pdf
- http://www.dauniv.ac.in/downloads/EmbsysRevEd_PPTs/Chap_7Lesson03EmsysNewTask.pdf
- Generador de NMEA : Description du périphérique NKE: interface sortie nmea.pdf
- DELANNOY Claude, - *Le Livre du C-*
- DRIX Ph., -*Langage C norme ANSI*-
- DOCUMENTACIÓN SKYWAVE : -*T11_Application_Controller_API_Reference*-, - *AN24_OBDII_Application*-